1.1 Introduction

1.1.1 Defining the Question

- Create a prediction model that more accurately predicts whether a user will click an Ad.

1.1.2 The Context

- A Kenyan entrepreneur has created an online cryptography course and would want to advertise it on her blog.
- She currently targets audiences originating from various countries.
- In the past, she ran ads to advertise a related course on the same blog and collected data in the process.
- She would now like to employ your services as a Data Science Consultant to create a solution that would allow her to determine whether ads targeted to audiences of certain characteristics i.e. city, male country, ad topic, etc. would click on her ads.

1.1.3 Metrics for Success

- Accuracy Score 85% or above.

1.1.4 Experimental Design Taken

- Installing packages and loading libraries needed
- Loading the data
- Exploratory Data Analysis
- Data Cleaning
- Visualizations
- Modelling: Random Forest
- Predictions and Evaluation of the Model
- Conclusion

1.1.5 Appropriateness of the Data

- Dataset link: link text
- The columns in the dataset include:
  - Daily Time Spent on Site
  - Age
  - Area Income
  - Daily Internet Usage
  - Ad Topic Line
  - City
  - Male
  - Country
  - Timestamp
  - Clicked on Ad

1.2 Installing & Loading Necessary Packages

Installing packages we need for the project analysis.

install.packages("iterators")
install.packages("caret") install.packages("caretEnsemble") install.packages("ggplot2") install.packages("e1071")
install.packages("randomForest") install.packages("ggcorrplot") install.packages('ranger') install.packages('caTools')
install.packages('rpart.plot')

```
# Importing Libraries we need for this Project analysis.

library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library(ggplot2)
library(lattice)
library(caret)
```

```
##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(rpart)
library(RColorBrewer)
library("rpart.plot")
```

1.3 Loading the Data

```
# Reading a csv file

adv <-fread("http://bit.ly/IPAdvertisingData")
```

```
# Viewing the top observations
head(adv)
```

```
##    Daily Time Spent on Site Age Area Income Daily Internet Usage
## 1:                    68.95  35    61833.90               256.09
## 2:                    80.23  31    68441.85               193.77
## 3:                    69.47  26    59785.94               236.50
## 4:                    74.15  29    54806.18               245.89
## 5:                    68.37  35    73889.99               225.58
## 6:                    59.99  23    59761.56               226.74
##                            Ad Topic Line           City Male    Country
## 1:    Cloned 5thgeneration orchestration    Wrightburgh    0    Tunisia
## 2:    Monitored national standardization      West Jodi    1      Nauru
## 3:      Organic bottom-line service-desk       Davidton    0 San Marino
## 4: Triple-buffered reciprocal time-frame West Terrifurt    1      Italy
## 5:         Robust logistical utilization   South Manuel    0    Iceland
## 6:       Sharable client-driven software      Jamieberg    1     Norway
##              Timestamp Clicked on Ad
## 1: 2016-03-27 00:53:11             0
## 2: 2016-04-04 01:39:02             0
## 3: 2016-03-13 20:35:42             0
## 4: 2016-01-10 02:31:19             0
## 5: 2016-06-03 03:36:18             0
## 6: 2016-05-19 14:30:17             0
```

```
# Viewing the bottom observations
tail(adv)
```

```
##    Daily Time Spent on Site Age Area Income Daily Internet Usage
## 1:                    43.70  28    63126.96               173.01
## 2:                    72.97  30    71384.57               208.58
## 3:                    51.30  45    67782.17               134.42
## 4:                    51.63  51    42415.72               120.37
## 5:                    55.55  19    41920.79               187.95
## 6:                    45.01  26    29875.80               178.35
##                           Ad Topic Line          City Male
## 1:        Front-line bifurcated ability  Nicholasland    0
## 2:        Fundamental modular algorithm     Duffystad    1
## 3:      Grass-roots cohesive monitoring   New Darlene    1
```

```
## 4:         Expanded intangible solution South Jessica    1
## 5: Proactive bandwidth-monitored policy   West Steven    0
## 6:      Virtual 5thgeneration emulation   Ronniemouth    0
##                   Country           Timestamp Clicked on Ad
## 1:                Mayotte 2016-04-04 03:57:48             1
## 2:                Lebanon 2016-02-11 21:49:00             1
## 3: Bosnia and Herzegovina 2016-04-22 02:07:01             1
## 4:               Mongolia 2016-02-01 17:24:57             1
## 5:              Guatemala 2016-03-24 02:35:54             0
## 6:                 Brazil 2016-06-03 21:43:21             1
```

```
# Checking the number of rows and columns
dim(adv)
```

```
## [1] 1000   10
```

```
# There are 1000 rows and 10 columns.
```

```
# checking the types of attributes (columns)

sapply(adv, class)
```

```
## $`Daily Time Spent on Site`
## [1] "numeric"
##
## $Age
## [1] "integer"
##
## $`Area Income`
## [1] "numeric"
##
## $`Daily Internet Usage`
## [1] "numeric"
##
## $`Ad Topic Line`
## [1] "character"
##
## $City
## [1] "character"
##
## $Male
## [1] "integer"
##
## $Country
## [1] "character"
##
## $Timestamp
## [1] "POSIXct" "POSIXt"
##
## $`Clicked on Ad`
## [1] "integer"
```

```
# checking the summary statistics of the dataset such as the mean

summary(adv)
```

```
##  Daily Time Spent on Site      Age          Area Income    Daily Internet Usage
##  Min.   :32.60            Min.   :19.00   Min.   :13996   Min.   :104.8
##  1st Qu.:51.36            1st Qu.:29.00   1st Qu.:47032   1st Qu.:138.8
##  Median :68.22            Median :35.00   Median :57012   Median :183.1
##  Mean   :65.00            Mean   :36.01   Mean   :55000   Mean   :180.0
##  3rd Qu.:78.55            3rd Qu.:42.00   3rd Qu.:65471   3rd Qu.:218.8
##  Max.   :91.43            Max.   :61.00   Max.   :79485   Max.   :270.0
##  Ad Topic Line          City                Male         Country
##  Length:1000        Length:1000        Min.   :0.000   Length:1000
##  Class :character   Class :character   1st Qu.:0.000   Class :character
##  Mode  :character   Mode  :character   Median :0.000   Mode  :character
##                                        Mean   :0.481
##                                        3rd Qu.:1.000
##                                        Max.   :1.000
##    Timestamp                       Clicked on Ad
##  Min.   :2016-01-01 02:52:10.00   Min.   :0.0
##  1st Qu.:2016-02-18 02:55:42.00   1st Qu.:0.0
##  Median :2016-04-07 17:27:29.50   Median :0.5
##  Mean   :2016-04-10 10:34:06.64   Mean   :0.5
##  3rd Qu.:2016-05-31 03:18:14.00   3rd Qu.:1.0
##  Max.   :2016-07-24 00:22:16.00   Max.   :1.0
```

```
# Summary information of the dataset
glimpse(adv)
```

```
## Rows: 1,000
## Columns: 10
## $ ‘Daily Time Spent on Site‘ <dbl> 68.95, 80.23, 69.47, 74.15, 68.37, 59.99, 8~
## $ Age                        <int> 35, 31, 26, 29, 35, 23, 33, 48, 30, 20, 49,~
## $ ‘Area Income‘              <dbl> 61833.90, 68441.85, 59785.94, 54806.18, 738~
## $ ‘Daily Internet Usage‘     <dbl> 256.09, 193.77, 236.50, 245.89, 225.58, 226~
## $ ‘Ad Topic Line‘            <chr> "Cloned 5thgeneration orchestration", "Moni~
## $ City                       <chr> "Wrightburgh", "West Jodi", "Davidton", "We~
## $ Male                       <int> 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0~
## $ Country                    <chr> "Tunisia", "Nauru", "San Marino", "Italy", ~
## $ Timestamp                  <dttm> 2016-03-27 00:53:11, 2016-04-04 01:39:02, ~
## $ ‘Clicked on Ad‘            <int> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1~
```

The glimpse output shows the datatypes of each column and a few observations.

1.5 Data Cleaning

Missing values

Completeness:

```
# Checking for missing values by columns

colSums(is.na(adv))
```

```
## Daily Time Spent on Site                     Age             Area Income
##                        0                       0                       0
##     Daily Internet Usage           Ad Topic Line                    City
##                        0                       0                       0
##                     Male                 Country               Timestamp
##                        0                       0                       0
##            Clicked on Ad
##                        0
```

```
# There are no missing values in the dataset from the output.
```

```
# Checking the column names
names(adv)
```

```
##  [1] "Daily Time Spent on Site" "Age"
##  [3] "Area Income"              "Daily Internet Usage"
##  [5] "Ad Topic Line"            "City"
##  [7] "Male"                     "Country"
##  [9] "Timestamp"                "Clicked on Ad"
```

```
# Replacing spaces in the columns names with an underscore
names(adv) <- gsub(" ", "_", names(adv))
```

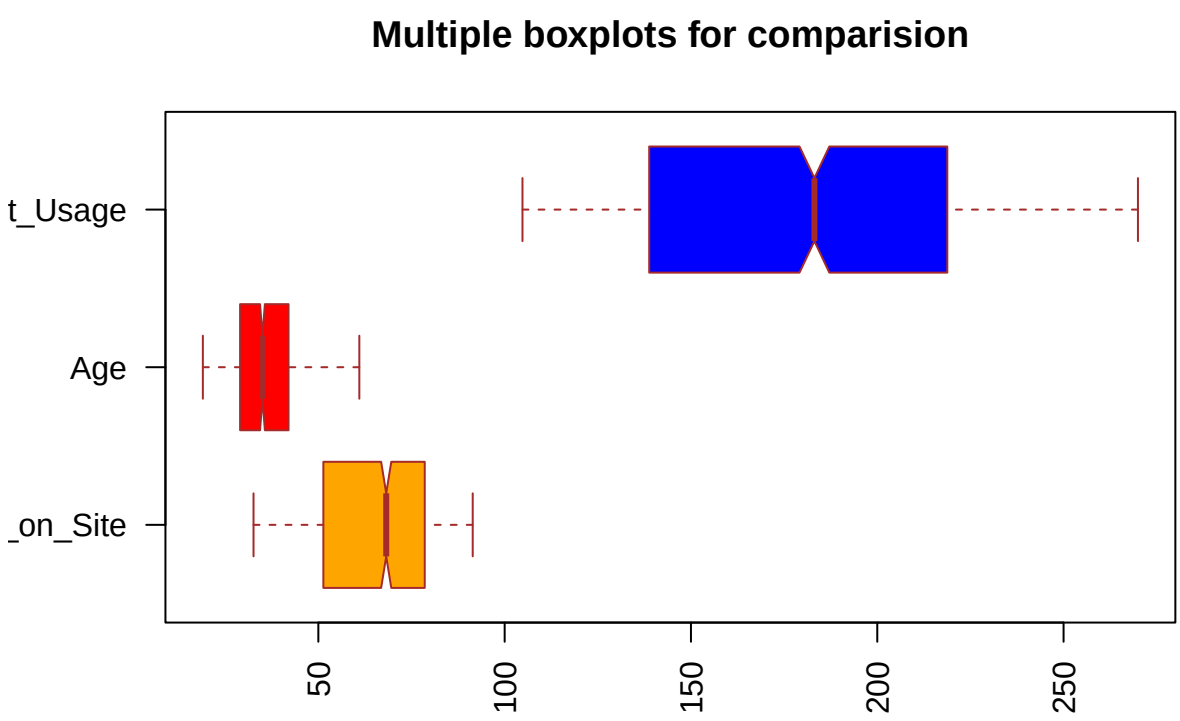
```
# Confirming the columns names have changed
names(adv)
```

```
##  [1] "Daily_Time_Spent_on_Site" "Age"
##  [3] "Area_Income"              "Daily_Internet_Usage"
##  [5] "Ad_Topic_Line"            "City"
##  [7] "Male"                     "Country"
##  [9] "Timestamp"                "Clicked_on_Ad"
```

Outliers

```
# Using a boxplot to check for observations far away from other data points.
# Using all three double type columns: specifying each
# labeling the title
```

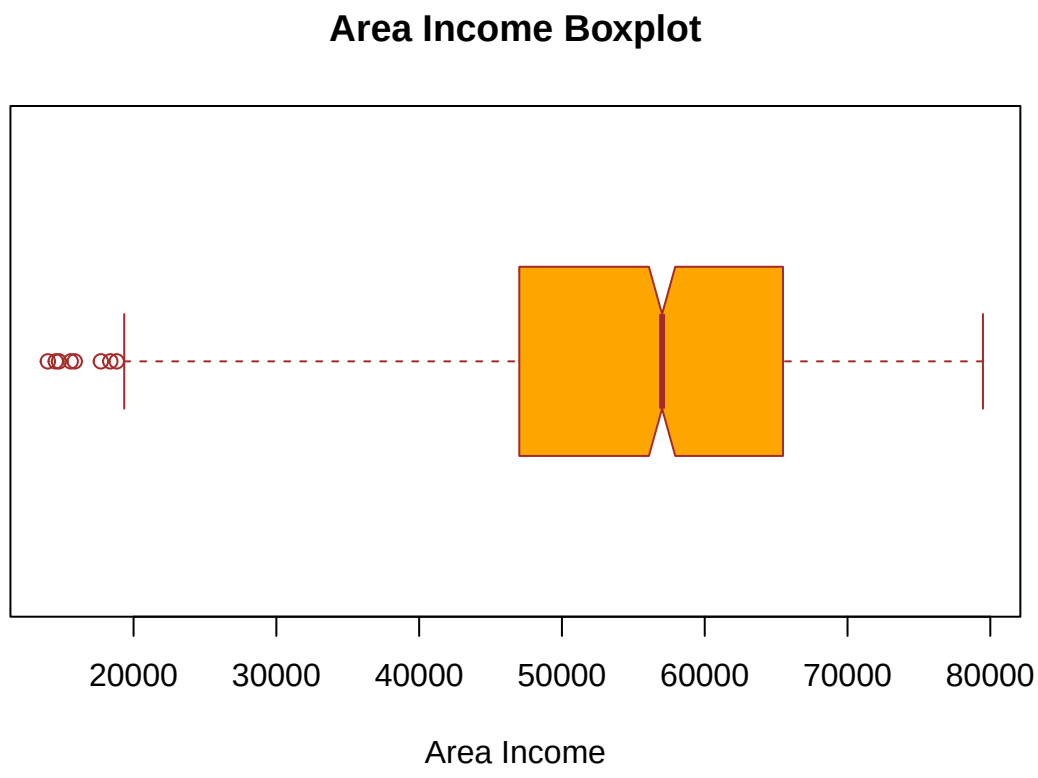
```
# labeling the x axis
# specifying color options

Daily_Time_Spent_on_Site <- adv$Daily_Time_Spent_on_Site
Age <- adv$Age
Daily_Internet_Usage <- adv$Daily_Internet_Usage


boxplot(Daily_Time_Spent_on_Site,Age, Daily_Internet_Usage,
main = "Multiple boxplots for comparision",
at = c(1,2,3),
names = c("Daily_Time_Spent_on_Site", "Age","Daily_Internet_Usage"),
las = 2,
col = c("orange","red","blue"),
border = "brown",
horizontal = TRUE,
notch = TRUE
)
```




```
# There are no outliers in the three features plotted
```

```
# Boxplot for the Area Income
# labeling the title
# labeling the x axis
# specifying color options
```

```
boxplot(adv$Area_Income,
main = "Area Income Boxplot",
xlab = "Area Income",
col = "orange",
border = "brown",
horizontal = TRUE,
notch = TRUE
)
```



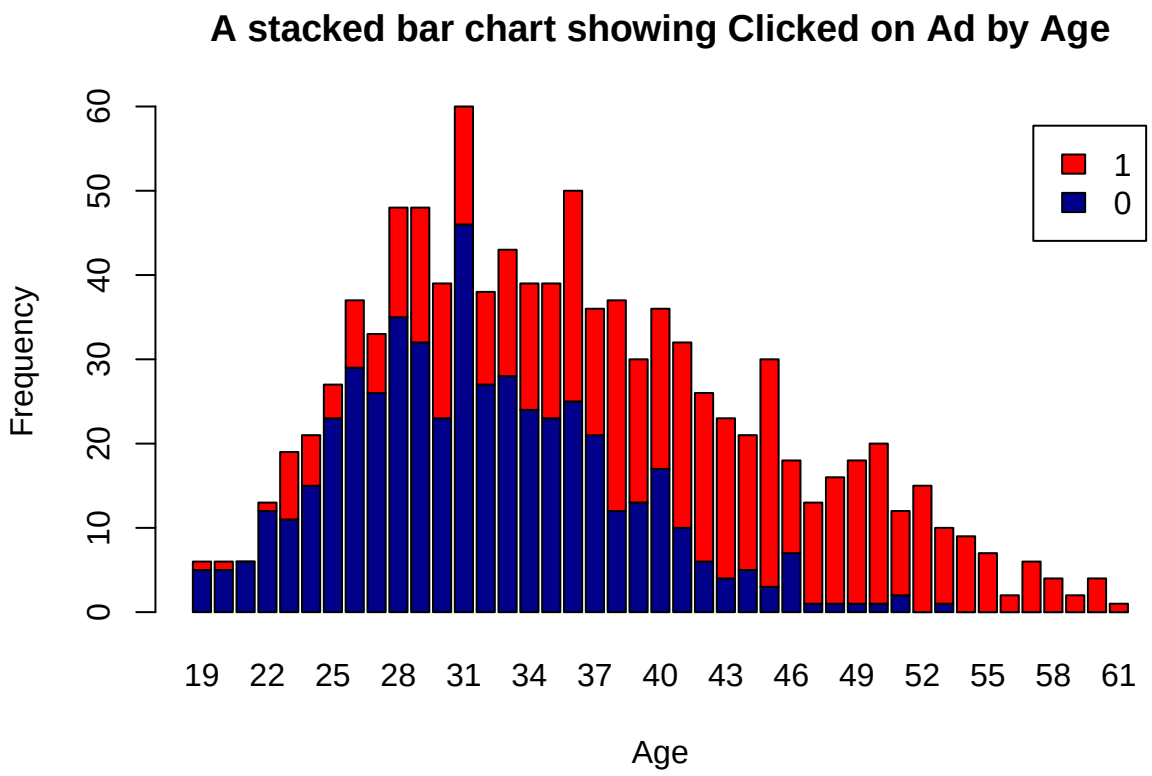

```
# There are a few outliers on the first quartile of the Area income boxplot.
```

1.6 Visualizations

Stacked bar chart

```
# Giving a title to the chart
# Labeling the x and y axis
# Setting the color options
# Creating a legend for easier reference
```

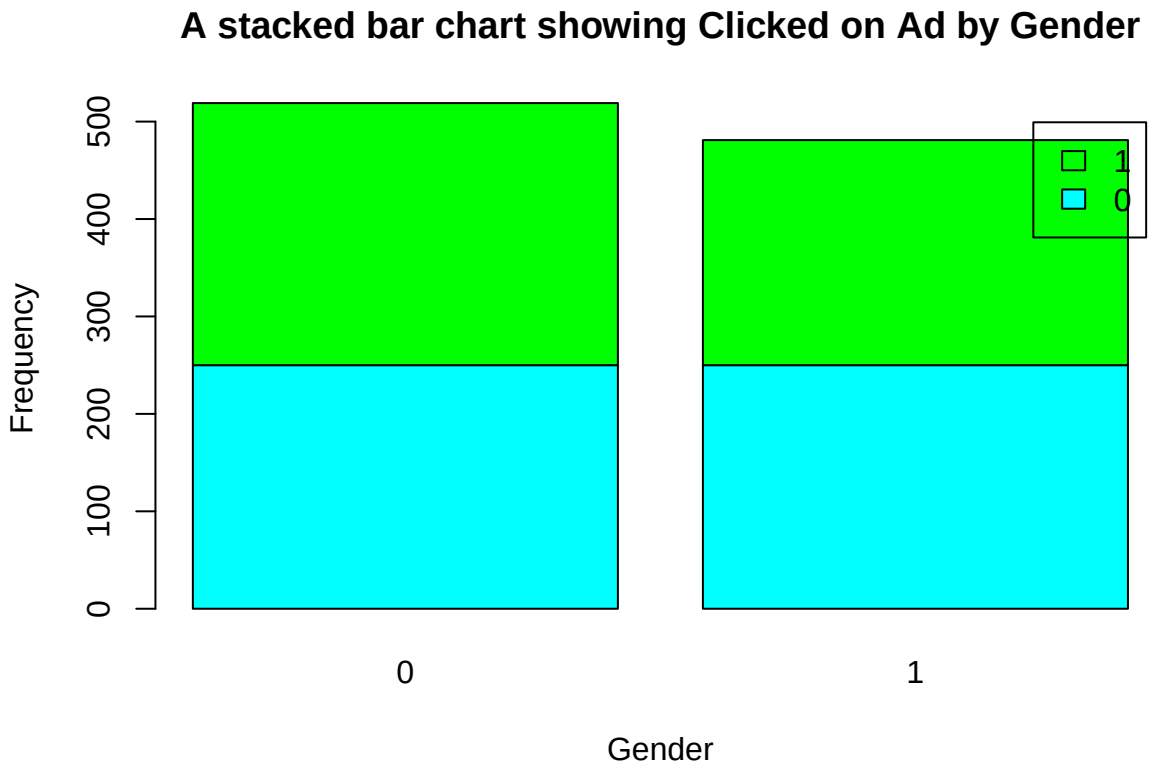
```
counts <- table(adv$Clicked_on_Ad, adv$Age)
barplot(counts,
  main="A stacked bar chart showing Clicked on Ad by Age",
  xlab="Age",
  ylab = "Frequency",
  col=c("darkblue","red"),
  legend = rownames(counts))
```



- 1 shows that the particpant clicked on an Ad.
- The stacked bar chart shows the distribution of the number of people who clicked on an Ad by age.
- The highest age of the participants was 61 and lowest was 19.
- The people who cliked most on Ads were between age 28 to 36.

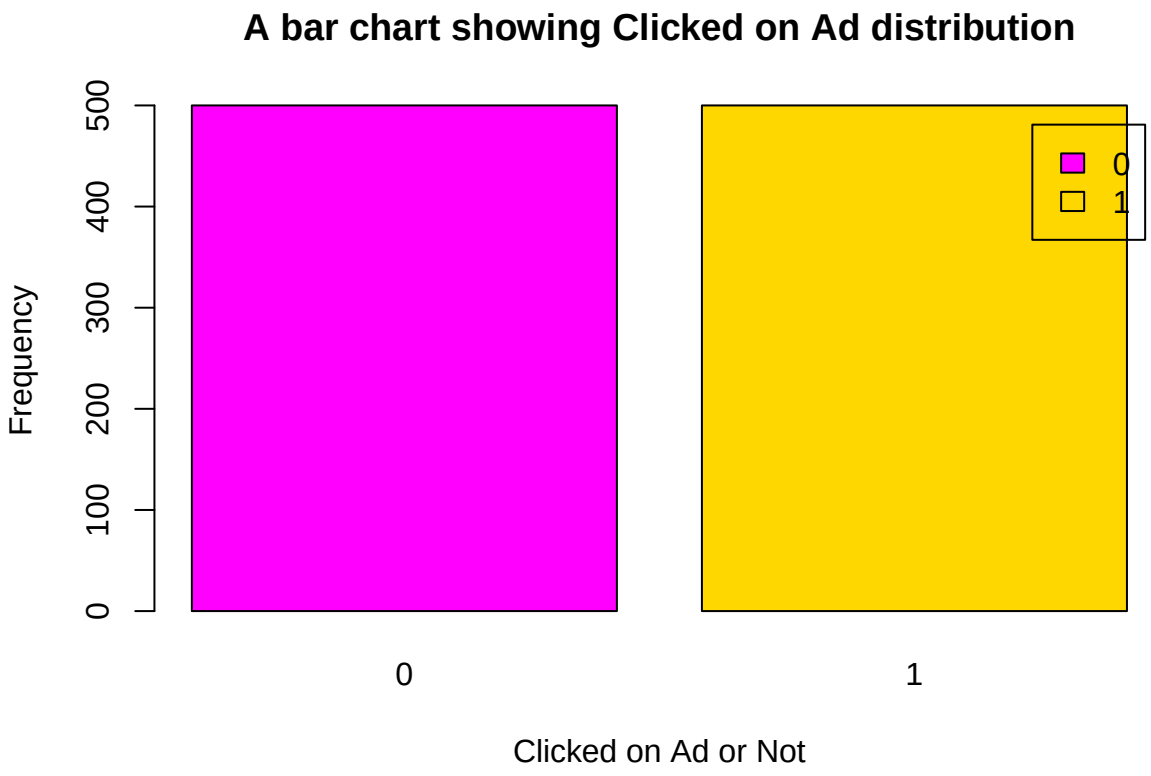
```
# Stacked bar chart
# Giving a title to the chart
# Labeling the x and y axis
# Setting the color options
# Creating a legend for easier reference

counts <- table(adv$Clicked_on_Ad, adv$Male)
barplot(counts,
  main="A stacked bar chart showing Clicked on Ad by Gender",
  xlab="Gender",
  ylab = "Frequency",
```

```
  col=c("cyan","green"),
  legend = rownames(counts))
```



- There are slightly more females than males in the dataset.
- More females clicked on Ad compared to males.

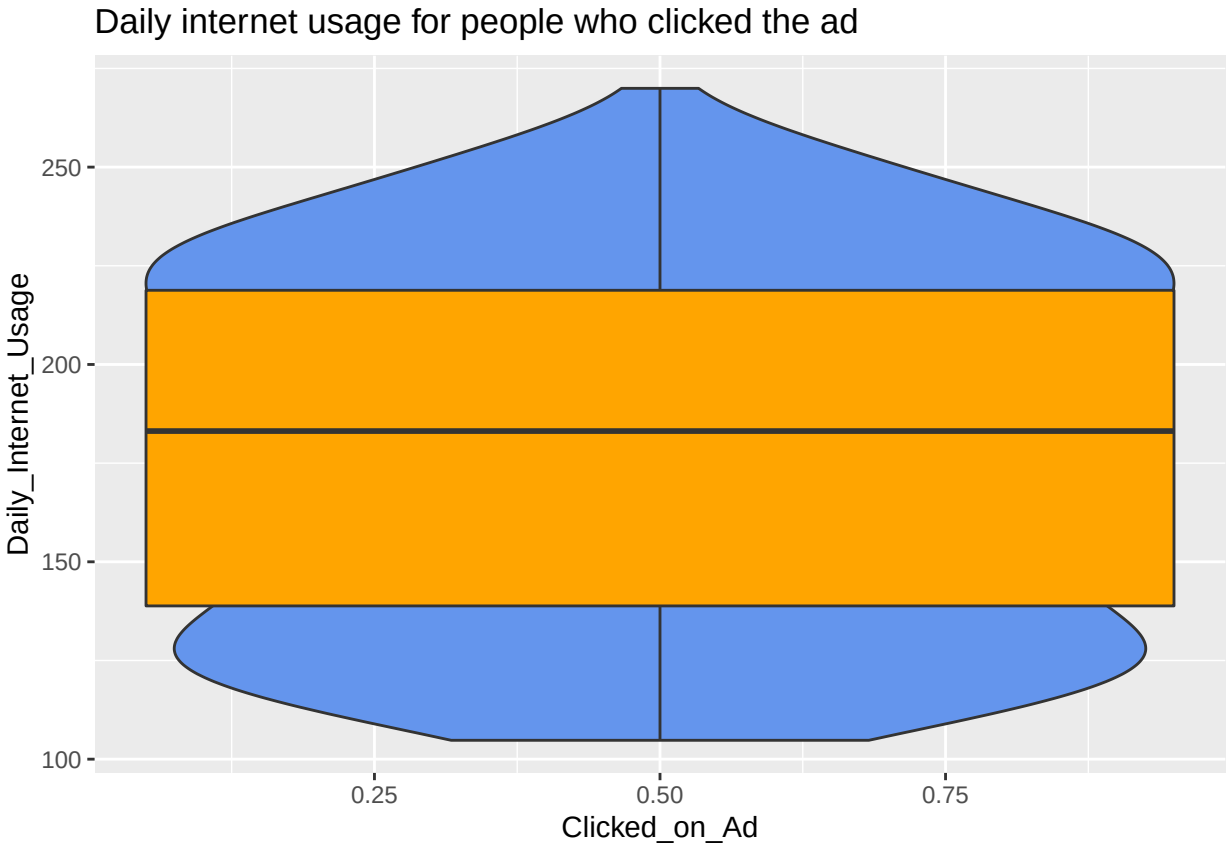
```
# Bar chart of the target variable

counts <- table(adv$Clicked_on_Ad)
barplot(counts,
  main="A bar chart showing Clicked on Ad distribution",
  xlab="Clicked on Ad or Not",
  ylab = "Frequency",
  col=c("magenta","gold"),
  legend = rownames(counts))
```



- The data is balanced since the number of people who clicked on Ad and those who did not are equal.

```
# A violin plot
# Specifying the x and y variables to be plot
# Setting the color
# Plotting a boxplot inside the violin plot
# Giving a title to the chart

ggplot(adv,
       aes(x = Clicked_on_Ad,
           y = Daily_Internet_Usage)) +
  geom_violin(fill = "cornflowerblue") +
  geom_boxplot(width = .2,
               fill = "orange",
               outlier.color = "orange",
               outlier.size = 2) +
  labs(title = "Daily internet usage for people who clicked the ad")
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



- People who click on Ad spend Daily internet amount between 135 and 220.
- There are no outliers.

1.7 Modelling

```
# Converting the target as a factor

adv$Clicked_on_Ad = factor(adv$Clicked_on_Ad, levels = c(0,1))
```

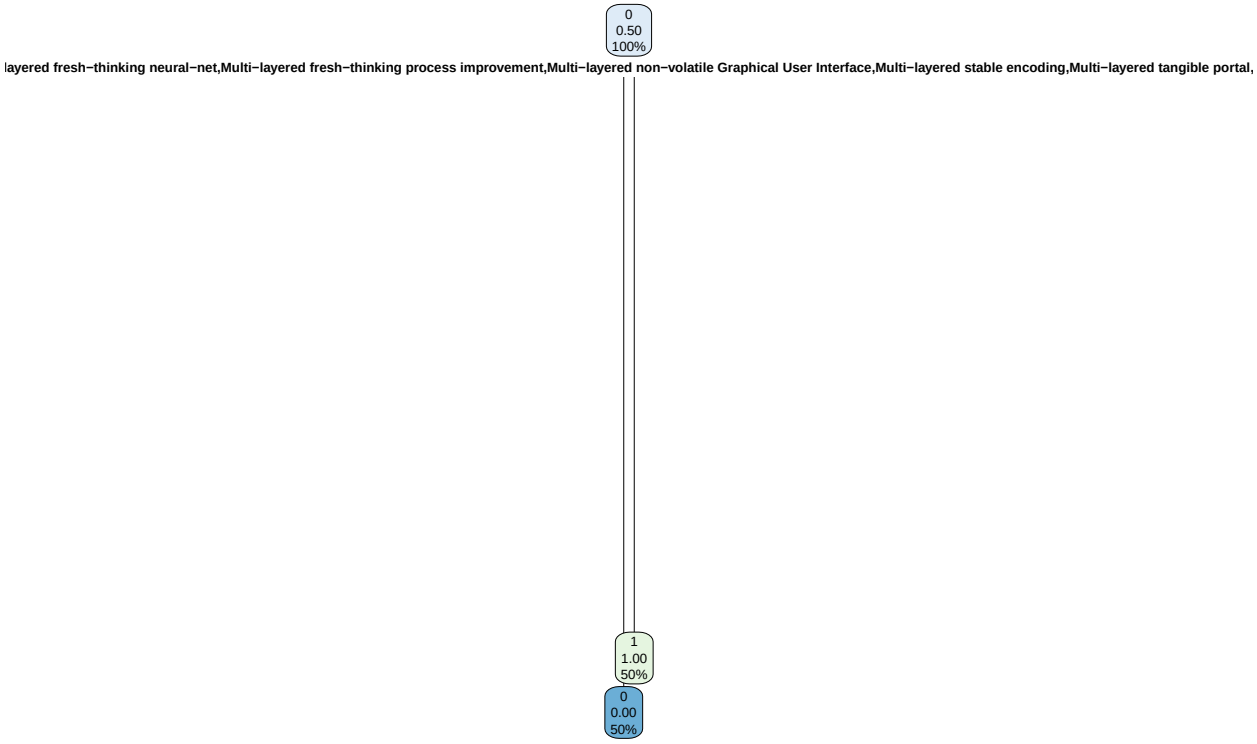
```
# checking the variable datatypes

sapply(adv, class)
```

```
## $Daily_Time_Spent_on_Site
## [1] "numeric"
##
## $Age
## [1] "integer"
##
## $Area_Income
## [1] "numeric"
##
```

```
## $Daily_Internet_Usage
## [1] "numeric"
##
## $Ad_Topic_Line
## [1] "character"
##
## $City
## [1] "character"
##
## $Male
## [1] "integer"
##
## $Country
## [1] "character"
##
## $Timestamp
## [1] "POSIXct" "POSIXt"
##
## $Clicked_on_Ad
## [1] "factor"
```

Decision Tree Classifier

```
# Using decision tree
# Fitting the model
# Specifying the target and predictor variables

m <- rpart(Clicked_on_Ad ~ . ,
    data = adv,
    method = "class")

# Plotting the decision tree model

rpart.plot(m)
```




```
# Making predictions
# Printing the confusion matrix

p <- predict(m, adv, type ="class")
table(p, adv$Clicked_on_Ad)
```

```
##
## p     0   1
##   0 500   0
##   1   0 500
```

```
# Printing the Accuracy

mean(adv$Clicked_on_Ad == p)
```

```
## [1] 1
```

- The model accuracy is 95.7%
- This is a good model to make predictions.
- We will evaluate this model or challenge it using another model.

1.8 Challenging the Solution

Random Forest Classifier

- The Random Forest model yielded an accuracy of 96.5%.
- This is a better model compared to Decision tree.

1.9 Conclusion

- Random Forest Classifier is the best model among the the two with an accuracy of 96.5%
- It is also better because it is a bagging method and uses many trees compared to decision tree classifier which uses only one tree.